Good for Enterprise™

GMM 8.3 EWS/SQL Deployment Planning Guide

Rev 1.5
Issue Date: 28-Jan-14
Last Updated: 17-June-15

Table of Contents

1 What's New

As this is the inaugural edition of the planning guide, everything in it is technically "new." Going forward, revised editions will use this space to outline any significant content added or changes made to the document, as well as to the product(s) and activities it describes.

Important: If you are replacing a currently deployed GMM 8.0 or GMM 8.1 with GMM 8.3, your concern is principally the installation (upgrade) of new software and preparations for high availability (HA) and failover. There is no MAPI to EWS move involved in upgrading to GMM 8.3. However, if you will be migrating users from GMM 7.*x* to GMM 8.3, there are additional steps involved, including upgrading your Exchange organization to at least an Exchange 2010 SP2 RU4 environment (Exchange 2013 CU2 is recommended), and SQL Server and EWS preparation, as well as preparations for HA and failover.

It is also important to bear in mind that when transitioning to Exchange 2010 or 2013 from an older version of Exchange, your Exchange environment will be in a "coexistence scenario" until deprecation of the legacy version. Such a scenario allows Exchange administrators to perform mailbox moves from the legacy platform to the new platform without a service disruption. After which, clients with deployments of GFE against Exchange 2007/2010 will need to utilize the new GMM 8.3 server version, which replaces MAPI with EWS for connecting to Exchange.

Consequently, and included as part of the planned migration of user mailboxes to Exchange 2013, the procedure in Section 5 – Deployment Steps should be followed to allow GFE-enabled users to continue synchronization without requiring a reprovisioning process.

2 Moving from MAPI to EWS

As alluded to in the previous section, in addition to aiding new customers planning their initial deployment of GMM, this document will help existing customers plan for and achieve the upgrade/migration of their existing GMM 7.*x*/8.0 site implementation to GMM 8.3/EWS with SQL Server. Its scope spans requirements gathering and assessment—including existing topology, scalability factors for accurate infrastructure sizing, and dependencies for optimized HA/DR. Step-by-step GMM 8.3 server installation and configuration instructions can be found in the *Good Mobile Messaging Administrator's Guide*. Appendix C herein presents PSR test results related to moving GFE users from a MAPI Exchange environment to an EWS Exchange environment.

2.1 Why the move to EWS?

For the typical enterprise Exchange environment, there are more than a few compelling reasons to migrate non-EWS applications. One is that Exchange Web Services (EWS) delivers a more versatile development environment than legacy APIs, extending the business logic from Outlook and Outlook Web Access to custom applications. Moreover, as organizations grow, the need for more customized software solutions grows increasingly important, with manageability becoming the central issue. To accommodate this evolutionary reality, EWS provides a single, cohesive API enabling email solutions and custom applications to interact with the Exchange store over HTTP.

In addition, GMM 8.3/EWS/SQL includes support for Exchange Online (aka Office 365).

2.2 GMM-MAPI versus GMM-EWS/SQL

Among the value-added differences between GMM 7.*x* and GMM 8.3 are:

- **Replacing MAPI with the EWS API for Exchange Connection, because…**
  - EWS is the preferred integration protocol for Exchange
  - EWS is more stable and reliable over higher latency networks
  - EWS supports Office 365

  - Microsoft has announced MAPI end of life (EOL) in April 2014.
- **Replacing file system storage with SQL Server-based storage, to attain…**
  - a familiar, resilient and reliable storage architecture
  - simplify storage management and planning.
- **Improved HA Solution, utilizing…**
  - a new "standby server pool" mechanism to provide HA for the GMM service
  - common SQL Server HA/DR solutions for data reliability.

The basic architectural differences are illustrated at a high level in Figure 1.

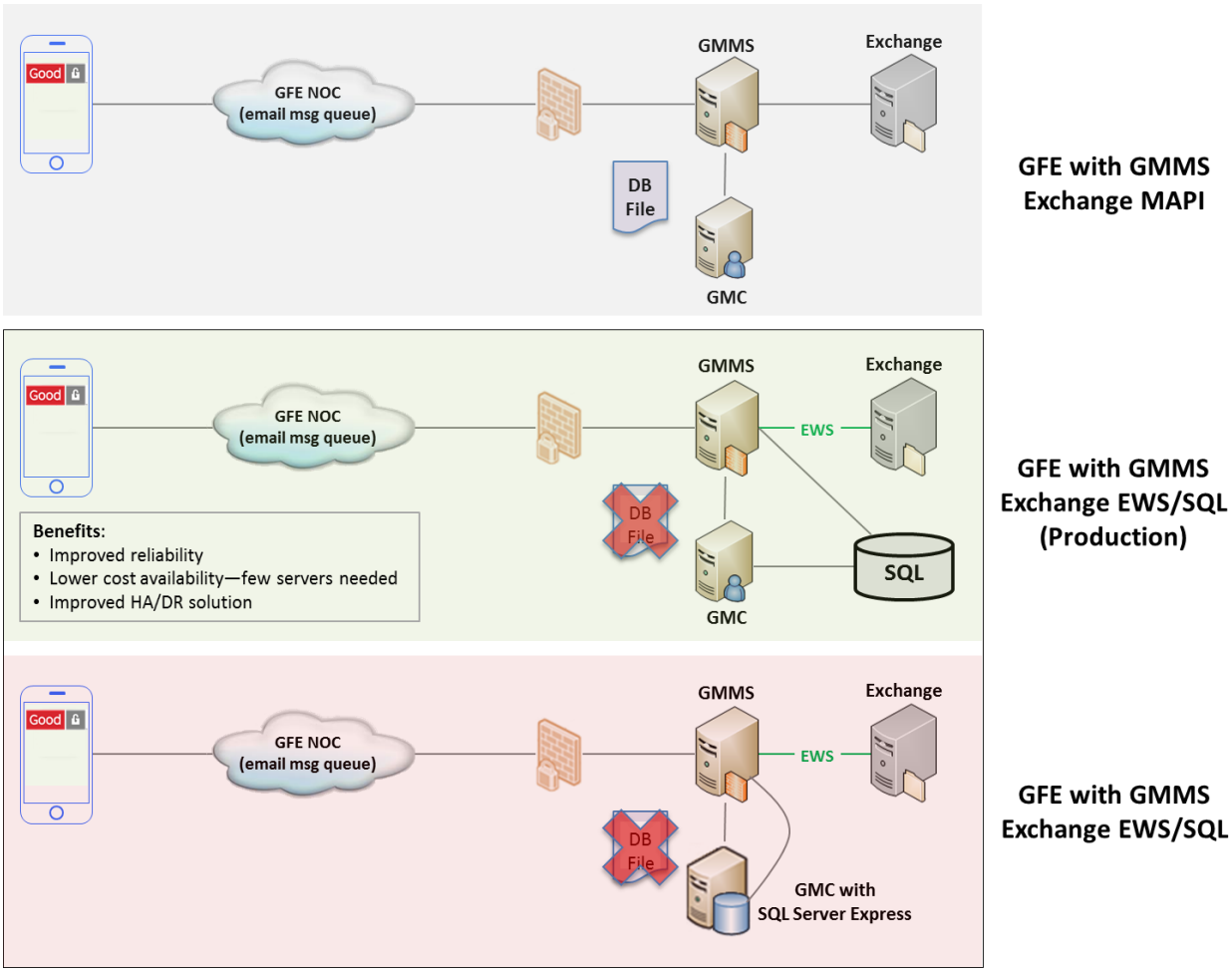


Figure 1: Replacing MAPI with EWS/SQL

SQL Server Express (pictured in the diagram at the bottom of Figure 1) is adequate for testing, and can even support a small production environment. Moreover, for very small deployments, GMC, GMM, and SQL Server Express can run on the same machine. Please note, however, that the features offered by SQL Server Express are not recommended for production environments of more than a few dozen users/devices. This is because the database has a 4 GB limit. Perhaps more importantly, HA and DR failover solutions are *not* supported by SQL Server Express. This includes log shipping.

3 Understanding and Scaling Your GMM-Exchange Infrastructure

Next, regardless of whether you're migrating from GMM 7.*x* to 8.3, upgrading from GMM 8.0 or GMM 8.1, or installing the Good Mobile Messaging server to implement GFE for the first time, it's important to understand the underlying architecture so you can appropriately scale your deployment to the specific needs of your enterprise.

This includes establishing your traffic and load profile with accurate measurements to adequately size the deployment in terms of the type and number of servers required, which will heavily influence the high availability and failover/recovery scenario you subsequently implement for your organization.

3.1 GFE Architectural Baseline

GMM 8.3 is the processing workhorse of GFE, synchronizing geographically dispersed mobile devices with your enterprise messaging servers; specifically, MS Exchange 2010 and later versions. GMM 8.3 uses a persistent cache to track the state of message synchronization. This Cache data is managed in a SQL database.

GMM 8.3 enables your IT administrator to set a wide variety of policies enforced on client devices. These include passwords, storage-card encryption, mandatory or permitted applications, S/MIME, and other policies. The essential system components requiring deployment in the enterprise include:

- **GFE Client** – software for containerizing enterprise data on mobile devices.
- **Good Mobile Control (GMC)** – server console controlling user and profile data stored in a SQL database, including secure access and application use policies.
- **Good Mobile Messaging (GMM)** – the processing engine of GFE for authenticating mobile device users and enforcing the enterprise security policies set by GMC.
- **MS Exchange** – your enterprise mail server, calendaring software and contact manager.
- **Microsoft SQL Server** (Enterprise or Standard) – database that stores the cache of data synchronized between each user's exchange mailbox and their handheld devices equipped with the GFE client.

Figure 2 offers a high-level glimpse of the GFE architectural baseline, which features GMM to Exchange-initiated communication with bi-directional traffic flow.

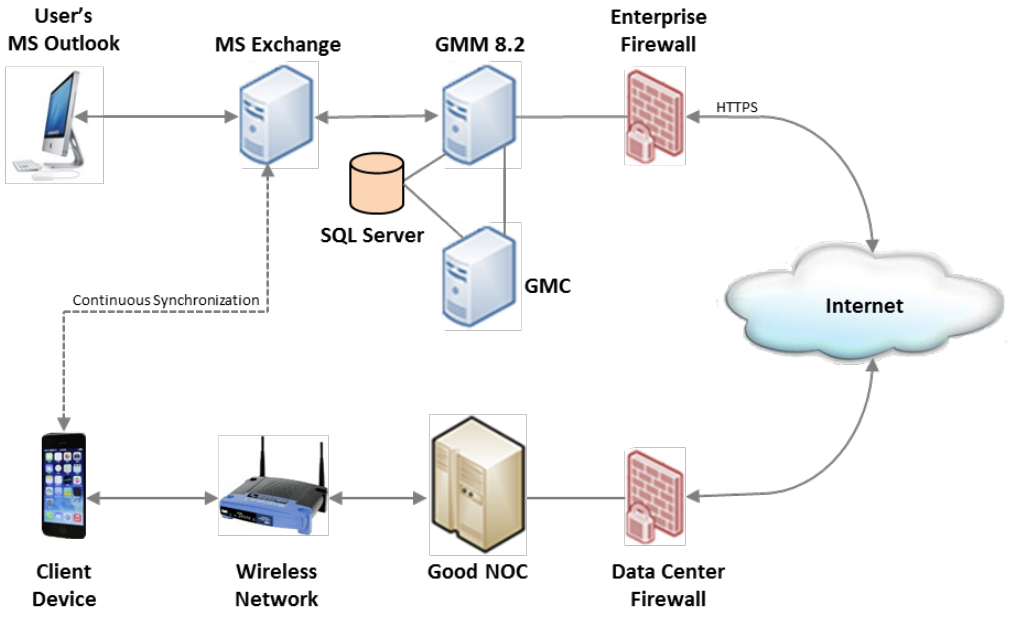


Figure 2: GMM to Exchange-Initiated Communication with Bi-directional Traffic Flow

To plan the optimal GMM 8.3 configuration for a new installation—that is, if you have never deployed an earlier GMM version in your enterprise—you will need to project a "normal" traffic and use profile to determine your everyday capacity requirements, and then increase these estimates for heavy traffic and load spikes.

Otherwise, if you are upgrading/migrating from GMM 7.*x* or 8.0, you have the convenience and better accuracy of measuring current "actual" traffic and use to determine your system capacity requirements before deploying the new GMM server. This is of great advantage in deriving a solution that appropriately scales. Of course, any discussion of scalability must necessarily be prefaced with an assessment of the factors driving system performance.

3.2 Scaling Factors

The scale of your GMM 8.3 deployment is largely dependent on the size of your enterprise and its IT logistics—number of sites, distance between sites, number and distribution of mobile users, traffic levels, latency tolerance, high availability (HA) requirements, and disaster recovery (DR) requirements.

With respect to HA/DR, two elements must be considered—applications and data. Most commonly, though not exclusively, HA refers to applications; i.e., GMM 8.3 and SQL Server. With clustering, there is a failover server for each primary server (2xN). DR focuses on both applications and data availability. The primary driver of your DR solution is the recovery time objective (RTO). RTO is the maximum time and minimum service level within which a business process must be restored after a disaster to avert an unacceptable break in business continuity.

Before contemplating the optimal number of servers to be deployed, however, it's wise to first determine the right size of an individual server to meet your enterprise's "normal use" profile. There are a number of methods for projecting a traffic and use profile. Actual, real-world measurement is recommended and made easy using built-in Windows Performance Monitoring tools. Notwithstanding the method applied, it is important to remember that GMM 8.3 server performance is governed by two principal factors: CPU utilization and available memory, the former being somewhat more critical than the latter.

3.2.1 CPU Utilization

CPU hardware and features are rapidly evolving, and your performance monitoring and analysis methodologies may need to evolve as well. Regardless, CPU utilization data is almost always useful. It is a piece of information that speaks to system performance. The real problem comes when you try to put one measurement in context by comparing it to another piece of data from a different system or a different test run. Not all CPU utilization measurements are comparable, even when two measurements are taken on the same make and model of processor. Even the tools used to measure performance can affect the validity of the comparison.

Monitoring results in Good's test environment over a three-day run yielded the CPU performance metrics included at the end of Appendix C.

3.2.1.1 Best Practices for Measuring Your CPU Utilization

Event Tracing for Windows (ETW) is a general-purpose, high-speed tracing facility provided by the operating system. Using a buffering and logging mechanism implemented in the kernel, ETW provides a tracing mechanism for events raised by both user-mode applications and kernel-mode device drivers. Additionally, ETW gives you the ability to enable and disable logging dynamically, making it easy to perform detailed tracing in production environments without requiring reboots or application restarts. The logging mechanism uses per-processor buffers that are written to disk by an asynchronous writer thread. This allows large-scale server applications to write events with minimum disturbance.

Otherwise, if you want to minimize the chances that hardware and OS features or measurement tools skew your utilization measurements, the following best practices[1] are advocated by Microsoft's Windows Server Performance Team:

- Use the "High Performance" power policy in Windows or disable power management features in the BIOS to avoid processor frequency changes that can interfere with performance analysis.
- Turn off simultaneous multithreading (SMT), overclocking, and other processor technologies that can affect the interpretation of CPU utilization metrics.
- Affinitize application threads to a core. This will enhance repeatability and reduce run-to-run variations. Affinitization masks can be specified programmatically, from the command line, or can be set using the GUI in Task Manager.
- Do NOT continue to measure in production indefinitely using this configuration. Whenever possible, strive to measure with all appropriate performance and power features enabled.
- Understand the system topology and where GMM is running on the server in terms of cores, packages, and nodes, particularly if GMM is not explicitly affinitized. Performance issues can suddenly appear in complex topologies. ETW and XPerf in the Windows Performance Toolkit can help you monitor this information.
- Rebooting will generally change where unaffinitized work is allocated to CPUs on a machine. This can make topology related performance issues reproduce intermittently. Reboot and measure again several times, or explicitly affinitize to specific cores and nodes to help flush out any issues related to system topology. This does not mean that the final implementation is required to use thread affinity, or that affinity should be used to work around potential issues; it just improves repeatability and clarity, lending better accuracy to your measurements.
- Use the right performance sampling tools for the job. If your sample sets will cover a long period of time, Perfmon counters may be acceptable. ETW generally samples system state more frequently and is correspondingly more precise than Perfmon, making it effective with shorter duration samples. Of course, depending on the number of ETW "hooks" enabled, you could wind up gathering significantly more data and your trace files may be large.

3.2.1.2 Creating a Data Collector Set from Performance Monitor

Real-time viewing of data collectors is just one way to use Performance Monitor. Once you have created a combination of data collectors that show you useful information about your system in real time, you can save them as a Data Collector Set, which is the building block of performance monitoring and reporting in Windows Reliability and Performance Monitor. It organizes multiple data collection points into a single component that can be used to review or log performance.

To create a Data Collector Set from Performance Monitor:

1. Make sure:
    - ***a*** You are logged on as a member of the local Administrators group, or you are logged on as a member of the Performance Log Users group and have completed the procedure to grant the **Log on as a batch job user** right to the Performance Log Users group.
    - ***b*** Be sure that the same user starting the GMM/GMC process is running the Data Collector.
    - ***c*** Windows Reliability and Performance Monitor service is running.
    - ***d*** At least one data collector is selected for display in Performance Monitor.
2. Right-click anywhere in the Performance Monitor display pane, point to **New**, and click **Data Collector Set**. The Create New Data Collector Set Wizard starts. The Data Collector Set created will contain all of the data collectors selected in the current Performance Monitor view.
3. Type a name for your Data Collector Set and click **Next**.

[1] http://blogs.technet.com/b/winserverperformance/archive/2009/08/06/interpreting-cpu-utilization-for-performance-analysis.aspx

4. The **Root Directory** will contain data collected by the Data Collector Set. Change this setting if you want to store your Data Collector Set data in a different location than the default. Browse to and select the directory, or type the directory name.

   Important: If you enter the directory name manually, you must not enter a back slash at the end of the directory name.

5. Click **Next** to define a user for the Data Collector Set to run as, or click **Finish** to save the current settings and exit.

6. After clicking **Next**, you can configure the Data Collector Set to run as a specific user. Click the **Change** button to enter the user name and password for a different user than the default listed.

   Important: If you are a member of the Performance Log Users group, you must configure Data Collector Sets that you create to run under your own credentials.

7. Click **Finish** to return to Windows Reliability and Performance Monitor, then…

   a. To view the properties of the Data Collector Set or make additional changes, select **Open properties for this data collector set**. You can get more information about the properties of Data Collector Sets by clicking the **Help** button in the Properties page.

   b. To start the Data Collector Set immediately (and begin saving data to the location specified in Step 4), click **Start this data collector set now**.

   c. To save the Data Collector Set without starting collection, click **Save and close**.

After you feel you've collected a reasonable sampling, go to the directory specified in Step 4 and assess the results.

3.2.1.3 Alternatives to Sampling

A straightforward alternative to periodically sampling the processor execution state is to measure the time spent in each processor state directly. This is accomplished by instrumenting the phase state transitions themselves. Processor state transitions in Windows are known as context switches. A context switch occurs in Windows whenever the processor switches the processor execution context to run a different thread. Processor state transitions also occur as a result of high priority Interrupt Service Routines (ISRs) gaining control following a device interrupt, as well as the Deferred Procedure Calls (DPCs) that ISRs schedule to complete the interrupt processing. By recording the time that each context switch occurs, it is possible to construct a complete and an accurate picture of CPU consumption.

3.2.1.4 Using XPerf to Analyze CSwitch Events

The same CPU busy calculations that Resource Manager makes can also be performed after the fact using the event data from ETW. This is the technique used in the Windows Performance Toolkit (WPT, better known as xperf), for example, to calculate CPU usage metrics.

Once you have downloaded and installed the Windows Performance Toolkit, you can launch a basic ETW collection session using the following xperf command:

```
xperf -on DiagEasy
```

Then, after you have accumulated enough data, issue another command to stop tracing and capture the event stream to a file:

```
xperf -d cputrace.etl
```

Next, process the **cputrace.etl** file using the xperfview app. After the trace file is loaded, xperfview provides visualizations that are very similar to Performance Monitor.

3.2.1.5 Interpreting CPU Performance Results

To summarize, the CPU utilization measurements at the system, process and thread level in Windows are based on a sampling methodology. Similarly, the processor queue length is also sampled. Like any sampling approach, the data gathered is subject to typical sampling errors, including:

- Accumulating an insufficient number of sample observations to be able to make a reliable statistical inference about the underlying population.
- Failing to ensure that faulty sources of sampling error are not causing sub-classes of the underlying population to be under- or over-sampled systematically.

So, these CPU measurements face familiar issues with regard to sampling size and the potential for systematic sampling bias, as well as the usual difficulty in ensuring that the sample data is actually representative of the underlying population (something known as non-sampling error). For example, the interpretation of the CPU utilization data that Perfmon gathers at the process and thread level is subject to limitations based on a small sample size for collection intervals less than, say, 15 seconds. At one minute intervals, there are enough samples to expect accuracy within 1–2%, a reasonable trade-off of precision against overhead. Over even longer measurement intervals, say 5 or 10 minutes, the current sampling approach leads to minimal sampling error, except in anomalous cases where there is some other source of systematic under-sampling of the processor's execution state.

Small sample size is also the reason that Windows does not currently permit Perfmon to gather performance data at intervals more frequent than once per second. Running performance data collection at intervals of 0.1 seconds, for example, the impact of relying on a very small number of processor execution state samples is quite evident. At 0.1 second intervals, processor times are calculated based on just 5 or 6 samples per interval. If you are running a micro-benchmark and want to access the same Thread\% Processor Time counters that Perfmon uses at 0.1 second intervals, you are looking for trouble. Under these circumstances, the % Processor Time measurements lose their resemblance to a continuous function over time.

CPU utilization consistently topping 80% of capacity is an indicator of overload and hence the need to scale up your system by increasing the number of cores or increasing the number of servers.

3.2.2 Memory Consumption

GMM servers are "large memory aware," allowing them to access more memory than normal 32-bit processes. This means the maximum memory consumption on Windows Server 2008 R2 is 4 GB per process. Earlier versions of Windows Server are not supported by GMM 8.3 EWS/SQL.

The recommended minimum configuration for GMM 8.3 is **4 Cores, 2.0 GHz or higher, 8 GB RAM**. Because you never want to see memory capacity at 100%, leave room for unexpected spikes in load. Ideally, set your ceiling at 85% of 4 GB of memory, or 3.4 GB RAM.

Good continues to refine and adjust system performance. Updated scalability and PSR sizing will be furnished in service releases. Existing customers who plan to upgrade/migrate to GMM 8.3 are encouraged to follow the guidelines set forth by Microsoft to determine actual usage in their specific environment. These can be found in http://technet.microsoft.com/en-us/library/cc749115.aspx for Windows Server and in http://msdn.microsoft.com/en-us/library/ms176018.aspx for SQL Server.

3.3 Additional GMM 8.3 Server Sizing Guidelines

The appropriate and adequate sizing of your GMM server(s) is largely a function of user demand and expected performance balanced against TCO.

As discussed above for GMM 8.3, general CPU and memory requirements comprise:

- 2–4 Core CPU; 4 cores recommended for higher loaded GMM servers.
- 2.0 GHz or higher
- 8 GB–12 GB RAM, with 12 GB recommended for higher loaded GMM servers.

Important: GMM is a 32-bit process and can only recognize a maximum of 4 GB RAM per process on Windows Server 2008 R2. GMM runs four processes, including **gdExchSyncSrv**, **GoodTech.GFe.EWSProc2**, **GdPushProc**, and **MemCached**.

Sizing and performance are not the same. Sizing is based on a conservative percentage of performance capabilities to allow for peaks and spikes in usage.

Even so, as a general rule, the following table outlines the maximum number of devices supported by one (1) GMM 8.3 server. And, by extension, the number supported by each additional GMM server.

Number of GMM Servers	Maximum Number of Devices
1	2,100
2	4,200
3	6,300
4	8,400
5	10,500
6	12,600

3.4 SQL Server Sizing Guidelines

GMM 8.3 requires a Microsoft SQL Server instance, which can be an existing Enterprise or Standard MS SQL Server 2008, 2008 R2, 2008 ENT, or 2012 instance already available within the organization. GMM can also connect to a remote SQL Server instance.

Multiple SQL Server instances can run on the same Windows Server, each of which can contain multiple databases. When multiple GMM servers are present, each must be assigned its own database. A SQL Server instance in this case is defined as a separate copy of SQL Server running on the same computer.

3.4.1 CPU/RAM

GMM servers can be supported across multiple SQL Servers, including those belonging to different clusters. However, Good recommends that each GMM Server installation point to a SQL instance running on a server with 4 cores and 8 GB memory. Thus, if a SQL Server will host multiple SQL instances, its CPU/RAM configuration must be increased proportionately.

This means that a SQL Server with two (2) database instances, each hosting a GMM server, will need:

(4 x 2 =) 8 cores and **(8 x 2 =) 16 GB of memory.**

Again, as indicated in Section 3.3 above, each GMM server can support 2100 devices when connected to a SQL Server that is 4-core and 8 GB, and a maximum of six (6) GMM databases can be supported per SQL Server. This means that the maximum number of devices that can be supported per SQL Server instance is 12,600, as shown in the following table.

Number of GMM Servers per SQL Server	Number of CPU cores per SQL Server	Memory (GB) per SQL Server	Number Of Devices Supported
GMM #1	4	8	2,100
GMM #2	8	16	4,200
GMM #3	12	24	6,300
GMM #4	16	32	8,400
GMM #5	20	40	10,500
GMM #6	24	48	12,600

3.4.2 Storage Capacity and IOPS Considerations

The following storage capacity sizing guidelines apply to SQL Server (≈60 MB per device on average):

Number of Devices	Storage Capacity
1,000	60GB
5,000	300GB
10,000	600GB

For best performance it is strongly recommended that Transaction Logs (not included in the above table) and the Database reside on two different physical disks. Transaction logs will grow at an average of 4.9 MB/user/average usage day; size of the logs will depend upon number of users and log rotation policy.

With regard to SQL Server Express, the foregoing suggests support for a maximum of 66 devices.

4 GB ÷ 60 MB ≈ 66

Another consideration in approaching the storage required is I/O per second (**IOPS**)—a common performance measurement used to benchmark computer storage devices like hard disk drives (HDD), solid state drives (SSD), and storage area networks (SAN). As with any benchmark, IOPS numbers published by storage device manufacturers do not guarantee real-world application performance. [2]

The most common performance characteristics measured are sequential and random operations. Sequential operations access locations on the storage device in a contiguous manner and are generally associated with large data transfer sizes, e.g., 128 KB. Random operations access locations on the storage device in a non-contiguous manner and are generally associated with small data transfer sizes, e.g., 4 KB. The performance characteristics can be summarized as follows:

Measurement	Description
Total IOPS	Total number of I/O operations per second (when performing a mix of read and write tests)
Random Read IOPS	Average number of random read I/O operations per second
Random Write IOPS	Average number of random write I/O operations per second
Sequential Read IOPS	Average number of sequential read I/O operations per second
Sequential Write IOPS	Average number of sequential write I/O operations per second

Figure 3 captures the disk I/O reading for SQLserv for a load with 2100 devices doing average email load and represents data captured during a 2-hour run window.

[2] Lowe, Scott (2010) "Calculate IOPS in a storage array". http://www.techrepublic.com; see also Atkin, Ian (2012) "Getting The Hang of IOPS v1.3", http://www.symantec.com/connect/articles/getting-hang-iops-v13.

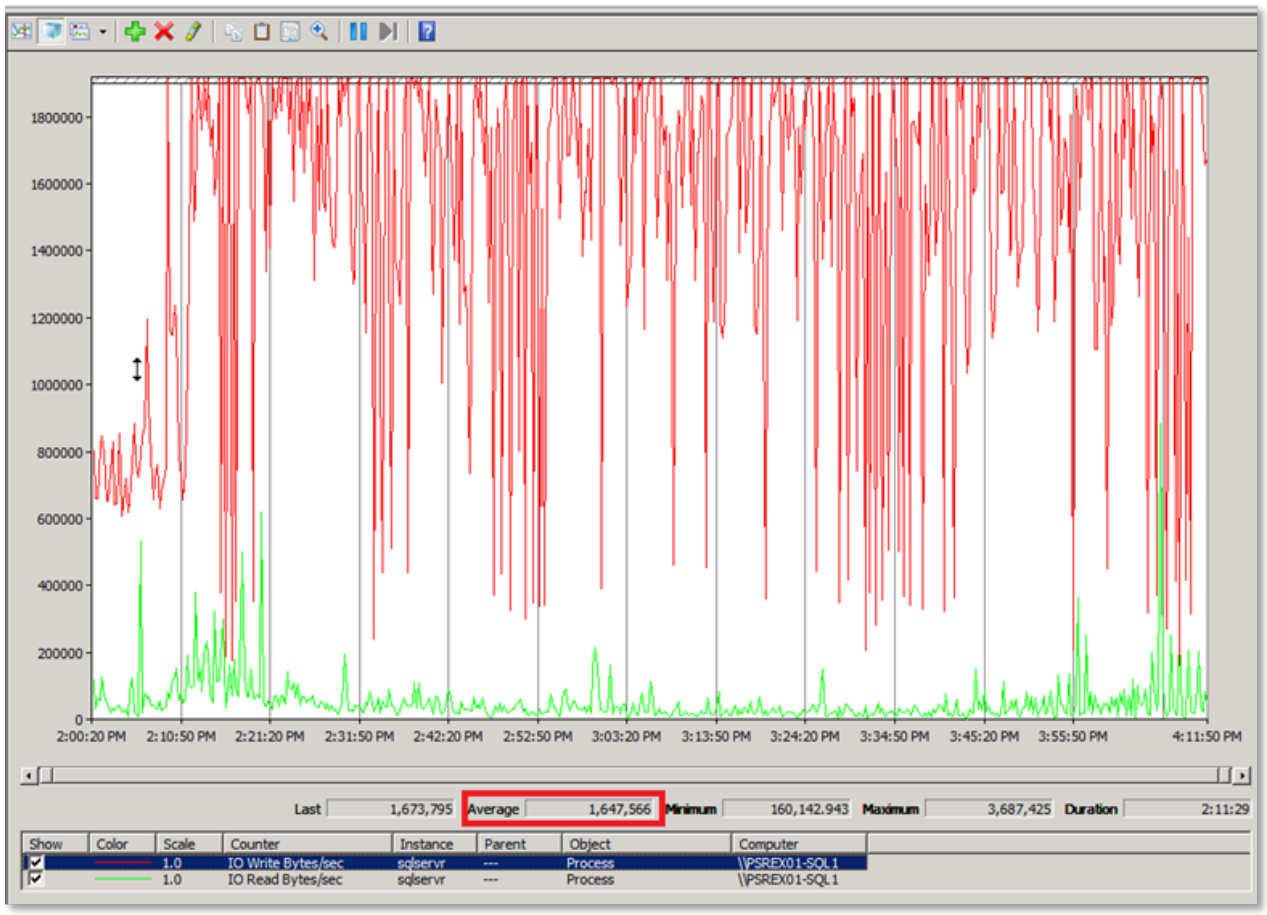


Figure 3: IOPS Result for GMM-SQL Server load of 2100 Devices Doing Average Email Load

As seen in the graph, these numbers are quite low. Even older SATA I drives can do 150MB/sec. Modern SATA III drives can do up to 600MB/sec. Because these SQLServ disk I/O numbers represent a mere 1–2% of server capability, GMMS-SQL disk I/O should not be a concern.

4 Migration Strategy and Approach

If you are currently using MAPI on Exchange 2010 or 2013, no Exchange migration is required to migrate GFE users to GMM 8.3 EWS/SQL. However, if your Exchange environment is earlier than 2010, there are important restrictions. Here are some tips to keep in mind in determining the best approach.

4.1 Exchange Migration

EWS/SQL only works with Exchange 2010 and newer versions. Hence, if you will be migrating your GFE users to GMM 8.3 while also migrating your Exchange environment up from earlier versions, remember that a direct move from Exchange 2003 to Exchange 2013 is not a coexistence scenario supported by Microsoft. Moreover, GFE using EWS is not supported against Exchange 2003 or Exchange 2007 by Good, which leaves the following three migration paths to choose from.

Scenario A: Enterprises using GFE with MAPI connectivity (GMM 7.*x*) against Exchange 2007 that are migrating user mailboxes to Exchange 2013.

Scenario B: Enterprises using GFE with MAPI connectivity (GMM 7.*x)* against Exchange 2007 that are migrating user mailboxes to Exchange 2010.

Scenario C: Enterprises using GFE with MAPI connectivity (GMM 7.*x*) against Exchange 2003 that are transitioning to Exchange 2010.

Additionally, if you are upgrading your existing Exchange 2010 organization to 2013, Microsoft advises[3] that you have a coexistence scenario only if both of the following conditions are met:

1. Exchange 2013 is deployed in an existing Exchange organization.
2. More than one version of Exchange provides messaging services to the organization.

Again, you cannot upgrade an existing Exchange 2003 organization directly to Exchange 2013. If you have an Exchange 2003 organization, Microsoft recommends moving from Exchange 2003 to Exchange 2010, and then upgrading from Exchange 2010 to Exchange 2013.

Warning: You must remove all instances of Exchange 2003 from your organization before you can upgrade to Exchange 2013.

The following table summarizes the scenarios in which coexistence between Exchange 2013 and earlier version of Exchange is supported.

Version	Exchange Organization Coexistence
Exchange 2003 and earlier	Not supported
Exchange 2007	• Exchange 2007 SP2 RU10 running on all Exchange 2007 servers in the organization, including Edge Transport • Exchange 2013 CU2 or later on all Exchange 2013 servers in the organization
Exchange 2010	• Exchange 2010 SP3 on all Exchange 2010 servers in the organization, including Edge Transport servers • Exchange 2013 CU2 or later on all Exchange 2013 servers in the organization
Mixed Exchange 2010 and Exchange 2007 organization	• Update Rollup 10 for Exchange 2007 SP3 on all Exchange 2007 servers in the organization, including Edge Transport servers (GMM 8.3 does not support Exchange 2007) • Exchange 2010 SP3 on all Exchange 2010 servers in the organization, including Edge Transport servers • Exchange 2013 CU2 or later on all Exchange 2013 servers in the organization

Please refer to your *Good Mobile Messaging Good Mobile Control for Microsoft Exchange Administrator's Guide* for additional GMM 8.3 installation prerequisites (hardware and software), then take the following steps in the order presented to assure successful migration and prevent your GFE users from experiencing synchronization issues on their devices.

4.1.1 GMM 8.3 Setup and Configuration

It is imperative that the change of GMM messaging server be done immediately after verifying the successful move of an Exchange user mailbox to the new Exchange Server version. Devices will not synchronize data post-Exchange mailbox migration until the appropriate steps have been taken to move the device to the new GMM-EWS server.

To prepare and configure the Exchange environment for GFE implementation:

1 Create a new **GoodAdmin** service account with mailbox on Exchange 2013.

2 Grant the **ApplicationImpersonation** permission to the new **GoodAdmin** service account.

3 Within the currently deployed GMC server, add the newly created service account as a Service Administrator from the **Roles** tab.

[3] http://technet.microsoft.com/en-us/library/jj898583(v=exchg.150).aspx

4. Using the newly created **GoodAdmin** account, install the GMM 8.3.*x* server on a new machine (in-place upgrade of GMM 7.*x* to GMM 8.3 is not supported) in accordance with the following provisos:
    a. Closely follow the complete installation instructions and recommendations found in your *Good Mobile Messaging Administrator's Guide*.
    b. Configure the server with the new **GoodAdmin** account as Local Administrator and logon as a service right granted to new **GoodAdmin**.
    c. During installation, configure the new GMM 8.3 server to use the same parent GMC server as the currently deployed GMM 7.*x* servers. Installing a second GMC server is not required. Adding the newly created account to the "Service Administrator" role in Step 3 above will allow the GMM server to authenticate to the GMC without being required to specify different credentials.
5. Verify that the environment is configured properly for the new GMM-EWS solution by adding a non-GFE-enabled user whose mailbox resides on Exchange 2013 and successfully completing the provisioning process.

    Note: Although this can be done using a currently enabled GFE user, it requires deleting the device within the GMC and then re-adding the user while selecting the newly installed GMM 8.3 server. Doing so also requires the end user to delete the application and reinstall it, subsequently reprovisioning with the newly provided 15-digit PIN.

As soon as you've verified that your environment is properly configured for this test case, you're ready to migrate your Exchange mailboxes and move your GFE devices to the GMM 8.3 server.

4.1.2 Moving Users to the New GMM Server

Take the following steps:

1. Within the GMC portal, initiate Change Messaging Server for the user device(s) associated with the Exchange user mailboxes moved.
2. If multiple mailboxes were migrated in batch, selection of the devices associated with these mailboxes may be done by selecting all of the associated devices and performing the Change Messaging Server option from the Apply Now dropdown box within the GMC.

    Once the change of messaging server is initialized within the GMC for migrating mailboxes, the users migrated will be presented with a popup informing them that the device must resynchronize with the its messaging server. The process can take up to 15 minutes depending on the size and number of items in a migrated user's Exchange mailbox.
3. Decommission the GMM 7.*x* server or upgrade it to 8.3 and reuse it later. See Chapter 10, "Uninstalling Good Mobile Messaging Server" in the *Good Mobile Messaging Administrator's Guide*.

Note: Good's Professional Services team is available to aid in streamlining mailbox moves for large customers migrating large numbers of users in single batches, as well as to consult and assist in any other aspects of your GMM 8.3 deployment. Visit http://www1.good.com/secure-mobility-solution/professional-services or contact your Good representative for more information.

4.2 Migration Paths

There are essentially two migration paths you can choose from, depending on the version of GMM you are currently running (before migrating). These migration paths include:

- MAPI-BDB (7.*x*) to EWS-SQL
- EWS-BDB (8.0.*x*) to EWS-SQL

Depending on the path appropriate to your situation, the migration can be a parallel server migration or an in-place upgrade.

4.2.1 Parallel Server Migration

Parallel server migration involves installing a new GMM 8.3 server in parallel with your existing environment (i.e., MAPI-BDB, EWS-BDB). Minimizing downtime is the primary benefit of this migration path, giving administrators the flexibility to select and schedule which users and devices to move to the new system and when. During this process, user data is recreated from the Exchange server on the new SQL DB. Users do not have to delete their app and get a new provisioning PIN, although when their account is moved, they will be notified (on the device) that their data must be recreated, whereupon they will need to tap or click the **OK** button to begin this procedure.

The user's email is rebuilt starting with the 500 latest emails and working backward. Their inbox and other synched folders will be synchronizing over time. Bookmarks stored within the Good secure browser are not deleted during the move. Also documents stored within the document repository are not deleted.

Moving users between MAPI and EWS Messaging Servers is fully described in the *Good Mobile Control Administrator's Guide* for GMC 2.4 and later versions.

Important: **Ensure that your users have installed a 2.4.*x* client or newer version before attempting to move them from MAPI to EWS SQL**.

4.2.2 In-place Upgrade

In-place migration involves upgrading directly from an earlier version of GMM to the latest version on the same server. During an in-place upgrade, devices are moved over to the new DB format automatically. While devices are in transition, they will not be able to receive or send mail.

The following in-place upgrade scenarios are supported:

- Upgrading from GMM 8.0 to GMM 8.3.0.18 or higher
- Upgrading from GMM 8.1 to GMM 8.3.18 or higher
- Upgrading from GMM 8.2 to GMM 8.3.18 or higher
    - As of 01/27/2014, version 8.3.0.18 can be obtained by contacting your Support Representative.

You can also do an in-place downgrade from GMM 8.3 to GMM 8.1, which involves reverting the database schema.

For complete pre-installation instructions and the steps to follow for an in-place upgrade, see *Upgrading Good Mobile Control and Migrating Good Messaging*.

5 Deployment Steps

The deployment checklist in Appendix A is designed to help you plan the specific details of your particular GMM 8.3 EWS/SQL deployment. Appendix B further offers a recommended implementation checklist of task items necessary to prepare your infrastructure, environment, and clients.

At a process level, however, deployment is accomplished by taking the following steps:

1. Research, define, and plan the high availability (HA) and disaster recovery (DR) solution appropriate to your enterprise and environment according to the options in Section 6 below.
2. Determine your traffic and use profile by one or more of the methods in Section 3.2.1.1, then acquire the appropriate number of GMM 8.3 machines required according to the capacity and configuration guidelines in Section 3.3.

3. Size, plan, and deploy the appropriate number of SQL Server machines/instances according to the guidelines in Section 3.4.
4. Set up and configure your GMM 8.3 servers according the instructions in Section 4.1.1.
5. Initialize new users or move existing users to GMM 8.3 according to the instructions in Section 4.1.2.
6. If moving users between MAPI and EWS messaging servers, follow the appropriate migration/upgrade path recommended in Section 4.2.

For complete instructions on configuring and executing an upgrade/migration to GMM 8.3 EWS/SQL, see *Upgrading Good Messaging and Good Mobile Control for Good Mobile Messaging™ Server Version 8.3.0 and Good Mobile Control™ Server 2.4.1 for Microsoft Windows Exchange*.

For comprehensive new system installation, migration/upgrade, and ongoing operations and system administration instructions see *Good Mobile Messaging Good Mobile Control for Microsoft Exchange Administrator's Guide* available from http://www1.good.com.

6 HA/DR Planning Considerations

When asked about HA/DR planning, many IT managers immediately think about data replication tactics such as tape backup, off-site vaulting, cloud backup, and remote data replication. As important as your organization's data is, however, it is only part of the broader requirement; namely, business continuity. Even if the data is recovered, it is virtually useless if the appropriate application is not up and running. Similarly, applications need to be protected from system failures, human error, and natural or man-made disasters.

At all events, you cannot manage what you cannot measure, so two planning elements are vital before anything else. The first is determining the hardware required to manage and deliver the IT services in question, the basis for which is outlined above in Section 3.2. Adequately allowing for growth, measuring as accurately as possible the number of devices, traffic and load likely to be placed on GMM 8.3 offers the best indication of the server hardware and supporting infrastructure likely to be required.

With particular respect to GMM 8.3 and its supporting architecture, the first objective in setting the goals of an HA/DR investment strategy is to develop a cost justification model for the expenses required to protect each component. If the expense exceeds the value the application and data provided to the business, plus the cost to recover it, then optimizing the protection architecture to reduce the expense associated with protection is an appropriate course of action.

6.1 RTO and RPO

Thus, for our purposes here, the first step in the HA/DR planning process is to balance the value of GFE and its supporting infrastructure against the cost required to protect it. This is done by setting a recovery objective. This recovery objective includes two principal measurements:

- **Recovery Time Objective (RTO)** – the duration of time and a service level within which the business process must be restored after a disaster (or disruption) to avoid unacceptable consequences associated with a break in business continuity. For instance, the RTO for a payroll function may be two days, whereas the RTO for mobile communications furnished by GMM to close a sale could be a matter of minutes.
- **Recovery Point Objective (RPO)** – the place in time (relative to the disaster) at which you plan to recover your data. Different business functions will and should have different recovery point objectives. RPO is expressed backward in time from the point of failure. Once defined, it specifies the minimum frequency with which backup copies must be made.

Obviously, if resources were fully abundant and/or free, then everything could have the best possible protection. Plainly, this is never the case. The intent of HA/DR planning is to ensure that available resources are allocated in an optimum fashion.

6.2 High Availability Alternatives

Availability is measured in terms of outages, which are periods of time when the system is not available to users. Your HA solution must provide as close to an immediate recovery point as possible while ensuring that the time of recovery is faster than a non-HA solution. Unlike with disaster recovery, where the entire system suffers an outage, your high availability solution can be customized to individual GFE resources; namely, GMC, GMM, and SQL Server.

Paradoxically, adding more components, whether primary or standby, to the system can actually undermine efforts to achieve high availability, mainly because complex systems have more potential failure points. Ideally, your HA solution will require less human intervention to restore system operation, the reason for this being that the most common cause for outages is human error.

Good therefore recommends distinct HA configurations for the primary service components comprising Good for Enterprise (GFE)—GMC, GMM, and SQL Server—with each discussed in turn next. Good's recommended disaster recovery solutions, being a breed apart from HA, are presented in Section 6.3.

6.2.1 GMC High Availability Model

Good Mobile Control (GMC) manages user, device, and profile data stored in a SQL database. Data changes are relatively infrequent and message flow is not impacted if GMC is offline. While clustering is a HA option, cold failover will typically meet most needs. It is also common for enterprises to have the standby server in their remote facility to provide for both failover and disaster recovery. Typically, clients do 24-hour logs during stable production, using the repair function in the GMC to identify any inconsistencies due to changes during the 24-hour window. During implementation, 15-minute logs will reduce rework while users/devices are being entered.

Because GMC is Microsoft Cluster aware, when GMC services are installed on each node, failover is automatic, as illustrated in Figure 4.

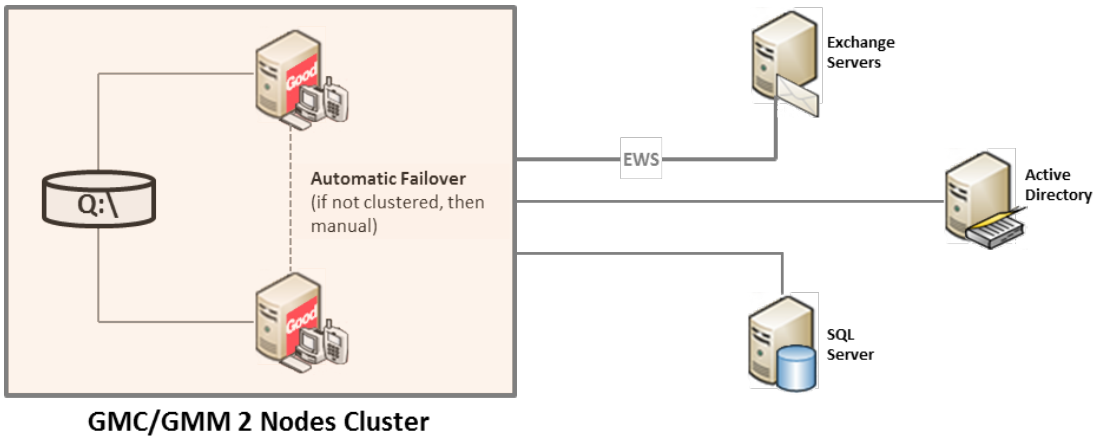


Figure 4: High Availability Model for GMC

Basically, GMC and GMM can run on the same host machine but cannot run on the same host machine as Microsoft Exchange Server. For deployments serving more than a thousand client devices, GMC and GMM should run on separate machines.

GMC should also be close to its SQL database to achieve latency of less than one millisecond. Greater latency will tend to slow the server. To optimize system performance, the SQL Server should also not be burdened with other work.

6.2.2 GMM 8.3 High Availability Model

Cold Failover is an N+1 HA design that is generally sufficient for smaller environments of less than 10 servers. For larger environments, an N+3 node configuration may be more appropriate, with one server used for rolling maintenance and updates, leaving two servers available for failover.

Essentially, Good recommends deploying one or more "spare" GMM servers. These spares can take over for any failed node, making MS clustering no longer required even while reducing downtime during failover.

Figure 5 shows this recommended Cold Failover model for most HA deployments.

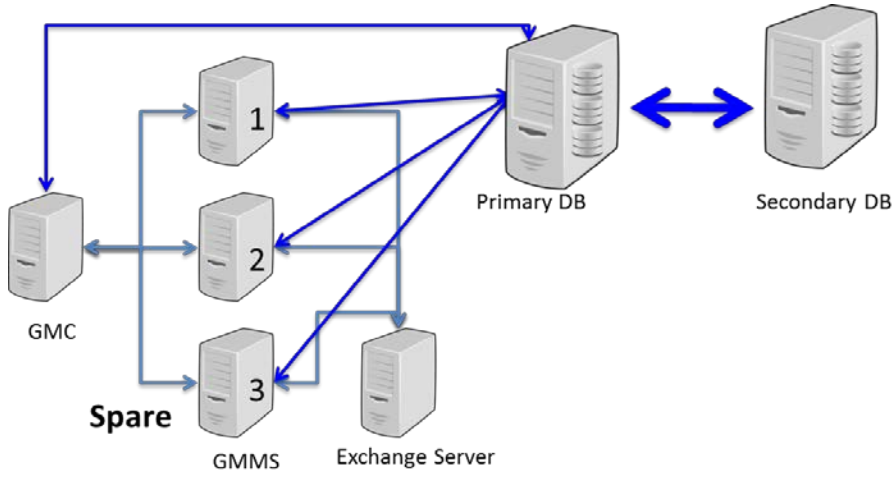


Figure 5: GMM-Exchange EWS/SQL High Availability Model

Obviously, the number of failover servers must be predicated on your special and/or unique business traffic, load , and risk profile, although this number will quite likely be much less than the number of primary GMM servers.

There must be a separate database for each GMM server. As previously pointed out, this database is the cache. Each spare or secondary GMM server is configured in "standby mode." On failover, you run the failover tool (**GMMFailoverTool.exe**) to have the standby take over the personality of the primary. The standby then pulls in all the configuration data of the primary from the database and connects to the NOC as if it were the original primary.

6.2.3 SQL Server HA Model

As pointed out previously, GMM maintains a cache of message synchronization state in the database. GMC also relies on a SQL database. Because of this, each secondary SQL Server database must be as close to a real-time data copy as possible or client devices will not remain in sync after failover.

Therefore, HA recommendations include:

- Clustering
- Mirroring (low latency required)
- AlwaysOn Availability Groups (SQL Server 2012)

Although database mirroring is currently a tested option in SQL Server 2008 R2 up to 2012, Microsoft has given notice that mirroring will be removed as a feature in a future version of SQL Server. Going forward, Microsoft advises using AlwaysOn Availability Groups.

6.2.3.1 Failover Clustering

Failover clustering is a collection of servers that by working together increase the availability of applications and services that run on the cluster. It can be described as a technology that automatically allows one physical server to take over the tasks and responsibilities of another physical server that has failed. A failover cluster provides high availability for cluster aware applications like GMC and SQL Server.

When you cluster SQL Server, you install one or more SQL Server instances into a Windows Failover Cluster. A Windows Failover Cluster uses shared storage. Typically, this shared storage is on a storage area network (SAN). The cost and complexity of SANs has dropped significantly in recent years to levels allowing wider adoption across both enterprise and small to medium size business environments.

When a SQL Server instance is installed on the cluster, system and user databases are required to be on the shared storage. This allows the cluster to move the SQL instance to any server (or "node") in the cluster whenever you request, or if one of the nodes is having a problem. There is only one copy of the data, but the network name and SQL Server service for the instance can be made active from any cluster node.

Figure 6 offers an example of a three-node failover cluster with SAN.

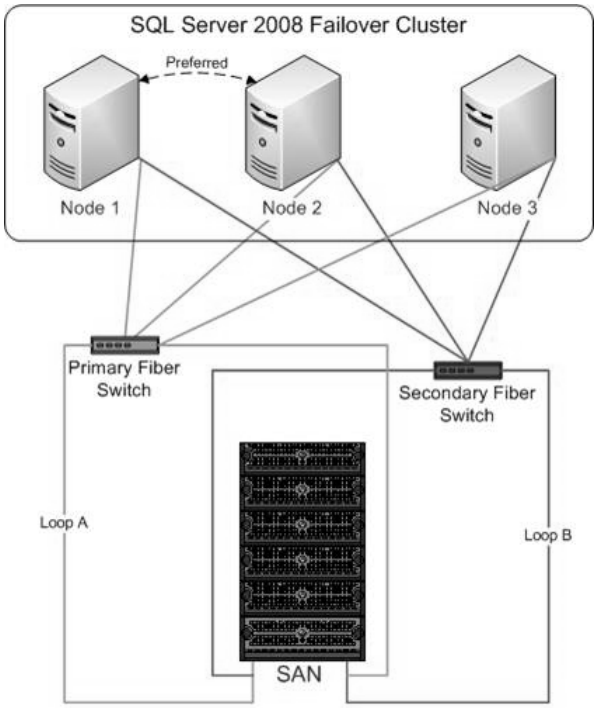


Figure 6: SQL Server (2008) Failover Cluster

Failover clustering does not do network load balancing and it will not improve scalability. It really only supports high availability. Moreover, clustering won't improve performance, unless you're moving to more powerful servers or faster storage at the same time you implement clustering. If you've been on local storage, don't assume that moving to a SAN is a panacea of performance. Also, clusters won't give you 100% uptime. There are periods of downtime when your SQL Server instance is failing over or moving between nodes.

For failover clustering in general, Microsoft recommends an N+1 topology modifying multiple-instance clustering where two or more nodes share the same failover node. The standby node requires significant hardware capabilities to support all N servers for the situations when they all fail simultaneously. However, N+1 uses resources effectively because there is only one standby node offline.

6.2.3.2 Mirroring

Database mirroring maintains two copies of a single database that must reside on different server instances of SQL Server Database Engine. Typically, these server instances reside on computers in different locations. Starting database mirroring on a database initiates a relationship, known as a database mirroring session, between these server instances.

One server instance serves the database to clients (the principal server). The other instance acts as a hot or warm standby server (the mirror server), depending on the configuration and state of the mirroring session. When a database mirroring session is synchronized, database mirroring provides a hot standby server that supports rapid failover without a loss of data from committed transactions. When the session is not synchronized, the mirror server is typically available as a warm standby server (with possible data loss).

All database mirroring sessions support only one principal server and one mirror server. Shown in Figure 7, this configuration further requires a Witness server to implement the automatic failover capabilities of GMM 8.3. The witness supports automatic failover by verifying that the principal server is up and functioning. The mirror server initiates automatic failover only if the mirror and the witness remain connected to each other after both have been disconnected from the principal server. Unlike the two partners, the witness does not serve the database.

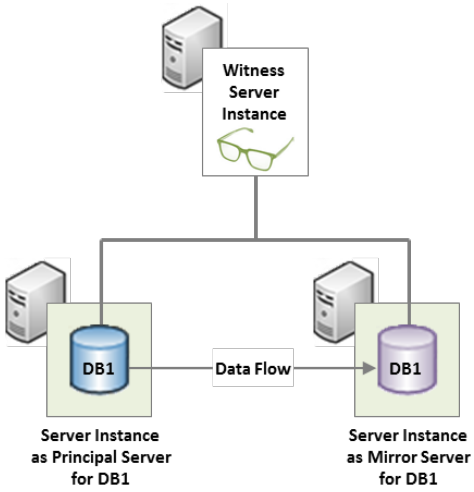


Figure 7: High-Safety Database Mirroring Session

GMM 8.3 supports synchronous mirroring only. Mirror servers must reside within the same data center as their primaries and, according to Microsoft, cannot have a network latency greater than 10 milliseconds between servers. Establishing a new mirroring session or adding a witness to an existing mirroring configuration requires that all involved server instances run the same version of SQL Server. However, when you are upgrading to SQL Server 2008 or a later version, the versions of the involved instances can vary.

6.2.3.3 AlwaysOn Availability

AlwaysOn is recommended by Good Technology and requires Windows Cluster, but not Quorum. Node Majority may be used. This requires three Windows servers:

- Prepare three Windows servers with the same configuration, Windows 2008 or 2012 .
- Prepare at least 25 GB disk space.
- Prepare an SQL Server 2012 setup file.

Refer to Chapter 8 of the *Good for Enterprise Administrator's Guide* for setup instructions.

6.3 Disaster Recovery Options

Your data is your most valuable asset for ensuring ongoing operations and business continuity. Disasters, unpredictable by nature, can strike anywhere at any time with little or no warning. Recovering both data and applications from one can be stressful, expensive and time consuming, particularly for those who have not taken the time to think ahead and prepare for such possibilities. However, when disaster strikes, those who have prepared and made recovery plans survive with comparatively minimal loss and/or disruption of productivity.

Of crucial importance is establishing a DR site separate from the primary site that a disaster could potentially strike. Good recommends mirroring your entire primary site configuration at the DR site, complete with the provision for synchronous byte-level replication of your SQL databases.

This is because if the system does fail, the replicated copy is up to date. To avoid a "User Resync" situation, the replica must also be highly protected.

To achieve this level of protection, files must be replicated using either:

- **Synchronous SAN (SRDF /S)** – Symmetrix Remote Data Facility (SRDF) is EMC^2 remote replication software for mission critical environments. "/S" is its synchronous option for zero data exposure loss.
- **Vision Solutions Double-Take** – allows real time, asynchronous replication at the byte level for either a physical, virtual or cloud environment, local or remote. This solution is supported only in conjunction with a Good Professional Services engagement.
- **Log Shipping** – fully incorporated in SQL Server Enterprise Edition, it uses a standby server that is not used during regular operations. A standby server is useful to help recover data if a disaster occurs. You can only use log shipping at the database level. You cannot use it at the instance level. However, if the primary server fails, you may lose the changes that were made by the transactions that occurred after your most recent transaction log backup.

While none of these three alternatives is the most expensive option available—fault tolerance, for instance, theoretically guarantees zero downtime in the event of full system failure and carries a commensurate cost—the costs to ensure responsive failover and recovery are not insignificant.

Therefore, before you commit to a disaster recovery solution, it's wise to look at each suggested alternative in detail, bearing in mind that, in general, the more spent, the greater the capability of a site and/or the entire enterprise to quickly resume operations after a disaster.

6.4 Good's Recommended DR Solution

No matter which replication product you choose, the replication software must have the ability to be installed with existing files, placing no limitation on file size. In addition, automatic SQL transaction replication of the database for GMC administration data in accordance with your business risk profile must also be supported in your finalized DR scenario.

In terms of data, Good recommends SAN replication, requiring low latency, although log shipping is a legitimate option—albeit with its own latency and bandwidth considerations—as well Vision Solutions Double-Take. AlwaysOn Availability is recommended.

Figure 8 shows the general DR model recommended by Good.

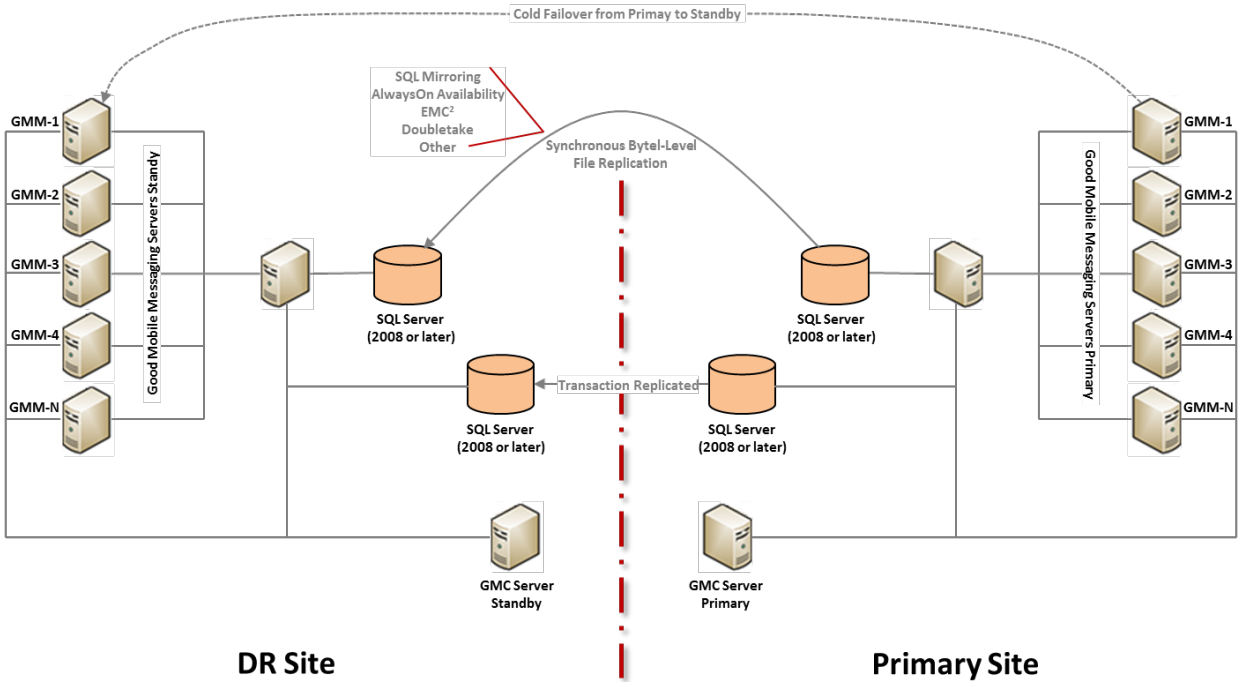


Figure 8: Recommended Disaster Recovery Model—N+1 Cold Failover from Primary to Standby

6.5 GMM 8.3 EWS/SQL – HA and DR Options Summary

The following comprise the supported HA and DR options for GMM 8.3.

MS SQL HA Options	**Notes:**
Windows Clustering	Requires additional servers
MS SQL Mirroring	Requires low latency (10ms or less)
MS SQL DR Options	**Notes:**
Synchronous SAN replication with EMC	Consult Good Professional Service
Asynchronous replication with Double-Take	Consult Good Professional Service
SQL Server Log Shipping	Requires Enterprise SQL
GMM Application HA Options	**Notes:**
Cold Failover with N+1 or N+N	Requires additional servers
GMM Application DR Options	**Notes:**
Cold Failover with N+1 or N+N	Requires additional servers

6.6 Comparative Failover and Recovery Timings

The bottom line in evaluating your various GMM 8.3 failover and recovery options and ensuring the solution aligns with the rest of your GFE system almost always boils down to “level of tolerance;” i.e., your tolerance for inconveniencing users. This is mainly because Exchange is the system of record and not data on the device. Still, the inconvenience created for the preponderance of your users can be reduced

with the right failover and recovery approaches. The relative recovery times for some representative failover options are shown in Figure 9.

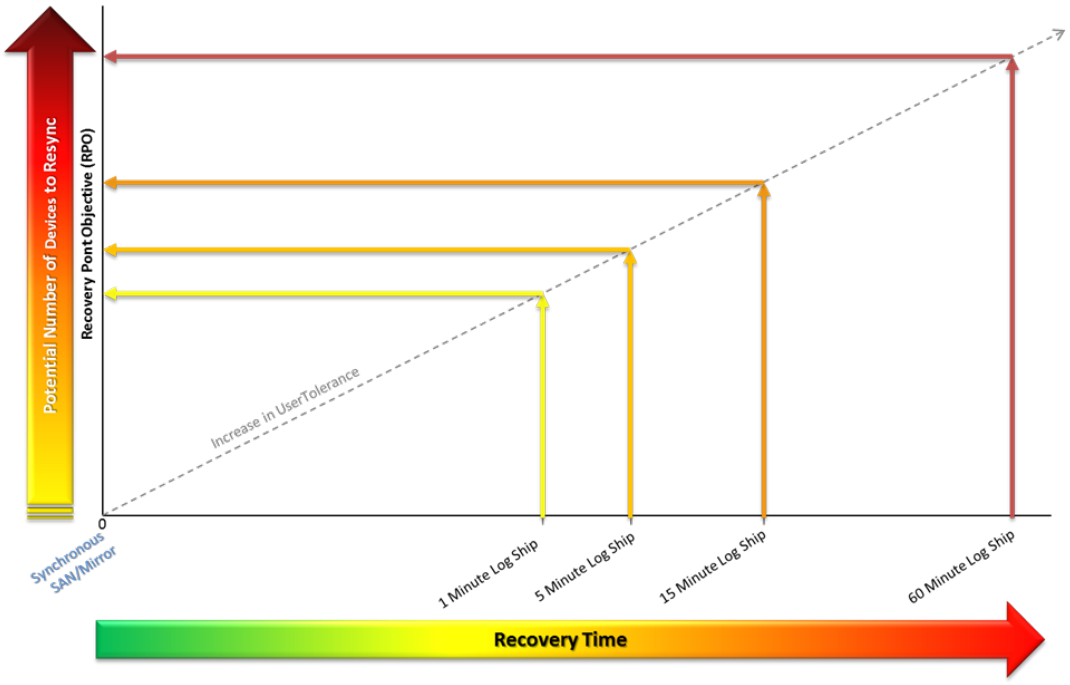


Figure 9: Recovery Time versus Number of Devices to Resync

As illustrated in Figure 9, RTO and RPO change significantly with the replication approach; recovery time increases in direct proportion to tolerance level, which is determined by the number of devices needing resync after a failover. The graph shows the failover solution correlated against the number of devices requiring resync, and, by extension, the number of users potentially inconvenienced. As a practical matter, an RPO of zero is only achievable with synchronous replication.

Replication Type	Distance	Bandwidth	RPO	Availability
Synchronous	Up to 150 miles	High bandwidth, available 100%	Zero RPO (replica is identical at all items)	100%

In the event of data loss on a device, GFE users can contact their administrator to initiate a RESYNC for their account from GMC. This will bring down 500 emails. If a greater number is desired, users can exercise a preference option to get an additional 500 until they have what they want.

Overall, synchronous replication requires a larger investment but delivers a RTO ranging from zero to as long as the “restart time.” Virtualization is another permutation, but the 100% availability for virtual servers is highly dependent on the availability of SAN running in the background on ESX servers and is only possible if SAN is available 100% of the time.

Direct PSR test results are included below.

Recovery Rate Test Data

PSR testing conducted by Good reveals the following recovery profiles for a system handling 2100 devices.

Data Type	*Failover Solution*				
	SQL Clustering	SQL Mirroring	Log Shipping		
			1 min Interval/ 50% users	5 Min Interval/ 50% users	15 min interval/ 100% users
Network Traffic (Pallet Size)	N/A	N/A	7 MB (w/ compression)	174 MB (no compression)	160 MB (no compression)
No. of Users (% disconnected)	0 (nil)	0 (nil)	70 (3.5%)	143 (7%)	825 (40%)
% Data Loss	0%	0%	See Result	See Result	See Result
Result	- no data loss - no user disconnects - no GMM server disconnect	- no data loss - no user disconnects - GMM server disconnects from primary SQL DB for a few seconds to reconnect to failover SQL DB	- less data loss - avg nwk traffic load - highest nwk traffic RT	- avg data loss - avg nwk traffic load - avg nwk traffic RT	- heavy data loss - high nwk traffic load - low nwk traffic RT

RT = Round Trips

Note: SQL mirroring requires the GMM server to point to both the primary and failover database. When failover occurs, GMM must disconnect from the primary database and reestablish connection with the failover database. Any significant disruption of service is avoided.

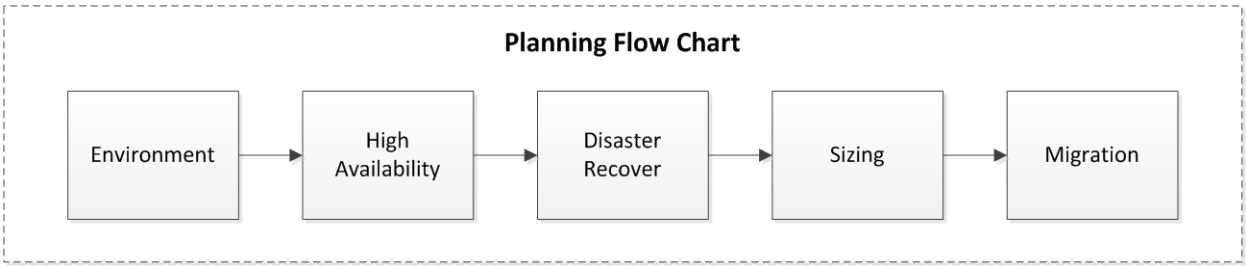
* * * * *

In conclusion, refer to *Upgrading Good Messaging and Good Mobile Control for Good Mobile Messaging™ Server Version 8.3.0 and Good Mobile Control™ Server 2.4.1 for Microsoft Windows Exchange* for detailed instructions on configuring and executing your upgrade/migration to GMM 8.3 EWS/SQL.

For comprehensive new system installation, migration/upgrade, and ongoing operations and system administration instructions see *Good Mobile Messaging Good Mobile Control for Microsoft Exchange Administrator's Guide* available from http://www1.good.com.

Appendix A: GMM 8.3 EWS/SQL – Deployment Planning Checklist

It is highly recommended that the following checklist be completed before deploying GMM EWS/SQL.

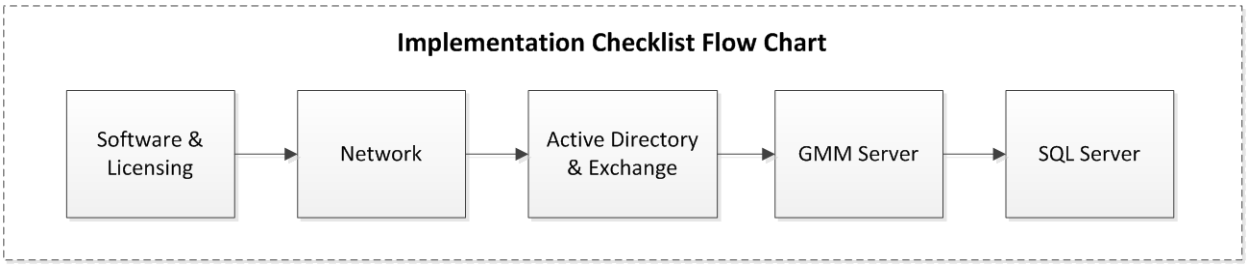


1	Environment	
	Check List	**Answer**
1.1	Microsoft Exchange Version	
1.2	Microsoft Exchange Locations (physical)	
1.3	Number of Exchange user mailboxes	
1.4	Number of handheld devices (estimate)	
2	**High Availability (HA)**	
	Check List	**Answer**
2.1	What is your Recovery Time Objective (RTO)?	
2.2	What is your GMM HA strategy?	
2.3	What is your SQL HA strategy?	
3	**Disaster Recovery (DR)**	
	Check List	**Answer**
3.1	What is your Recovery Point Objective (RPO)?	
3.2	Where are your DR locations?	
3.3	What is your GMM DR strategy?	
3.4	What is your SQL DR strategy?	

4	Sizing	
	Check List	**Answer**
4.1	How much SQL storage does your system require?	
4.2	How much CPU & RAM for SQL?	
4.3	How much CPU & RAM per GMM server?	
4.4	How many GMM servers?	
5	**Migration**	
	Check List	**Answer**
5.1	Are you migrating from an existing GMM environment?	
5.2	If so, what type of GMM is in use today?	
5.3	What is your migration strategy?	

Appendix B: GMM 8.3 EWS/SQL – Implementation Checklist

It is highly recommended that the following check list be completed before implementation takes place. This is a high level check list. For detail instructions, please reference the Administration Guide.



	Software and Licensing	Check
1.1	Download the correct software (should be 8.3.*x*)	☐
1.2	Make sure you have the appropriate license key (same licensing requirements as 7.*x* and 8.0.*x*)	☐

	Network	Check
2.1	Ensure that the GMM server has outbound (egress) access to the Good NOC on TCP port 443. The Good NOC has the following IP ranges: 216.136.156.64/27 198.76.161.0/24 To test appropriate access, open the following URL's on your GFE server – successful connectivity is noted by a "Congratulations!" message at the top of the page https://xml29.good.com https://xml28.good.com	☐
2.2	If the GMM server requires a Proxy server for external access. Please note the Proxy server information.	☐
2.3	If an on-site corporate wireless network is used, verify the following egress ports are open on your wireless network: Open egress to the Good NOC: UDP 12000	☐

	Network	Check
	TCP 15000 Open egress to Apple's APN (17.0.0.0/8): TCP 5223	

	Active Directory and Exchange	Check
3.1	Verify that your Exchange environment is supported: http://www1.good.com/support/compatibility-matrices.html	☐
3.2	Create an AD account for Good. The preferred user id is GoodAdmin • GoodAdmin user/password must not contain ':','@',or'/' characters • Password Expired option must be set to Never for this account. • GoodAdmin should be a member of Domain Users ONLY – NO other Groups.	☐
3.3	Create an Exchange mailbox for the GoodAdmin account. If this is an O365 deployment, make sure the GoodAdmin mailbox is then migrated to the O365 cloud (an enabled).	☐
3.4	Grant Application Impersonation permissions to the GoodAdmin account in Exchange (**very important!**).	☐
3.5	Ensure that your Exchange Autodiscover is setup correctly (**very important!**)	☐

	GMM Server	Check
4.1	Verify that you've provisioned the correct number of servers to support your deployment (including HA/DR).	☐
4.2	Verify that your server OS is supported: http://www1.good.com/support/compatibility-matrices.html	☐
4.3	Ensure that the GoodAdmin account is a local administrator on the server	☐
4.4	Ensure that the GoodAdmin account has "Logon As a Service" right	☐
4.5	Ensure that the server's time/date are set correctly	☐
4.6	Ensure that the server's has been joined to the domain	☐
4.7	Ensure that the windows firewall is off	☐

	GMM Server	Check
4.8	Ensure Antivirus/backup and backup software are stopped during the install	☐
4.9	Ensure that Microsoft .NET 3.5.1 is installed. This is a Windows feature, which can be added from windows.	☐
4.10	Ensure connectivity to your SQL server (usually TCP port 1433)	☐
4.11	Ensure connectivity to the GMC server	☐
4.12	Ensure connectivity to Exchange (EWS)	☐

	SQL Server	Check
5.1	Verify that your SQL version is supported: http://www1.good.com/support/compatibility-matrices.html	☐
5.2	Ensure that the GoodAdmin account has "dbcreator" permissions.	☐
5.3	Ensure SQL is configured properly for HA/DR (depends on your HA/DR options)	☐
5.4	If HA/DR is used, ensure the GoodAdmin account (or the SQL account used for failover) also have the following permissions: VIEW SERVER STATE ALTER ANY CONNECTION	☐

Appendix C: PSR Test Results – Moving Users (MAPI to EWS)

This appendix describes the process and results for the Good Mobile Messaging Server — Move Users (MAPI to EWS) performance test. The goal of the test was to measure the impact of initial synching of mass devices on GMMS/Exchange. The test was executed on the PSR-EWS-02 testbed.

Load Specification

Test set-up comprised two phases. Phase 1 involved setup and initialization of users/mailboxes. Prior to the test run, users' mailboxes were pre-populated with roughly 100MB of data, as follows:

- Inbox: 1500 messages.
- Sub-folders 200 messages each
- 38 Calendar Events
- 45 Contacts

After the mailbox database was initialized, the client simulators were placed online, ready to communicate with GMMS, and then LoadGen was started to generate traffic. The Microsoft LoadGen tool was used to simulate users interacting with Exchange Server via an Outlook client. All users were simulated using LoadGen's Light profile which executes 178 varied tasks per day per user. In addition to the default Very Heavy profile actions, simulated users were configured to send replies and forwards for a percentage of messages received.

The following is a summary of the key LoadGen parameters:

- Create 3 emails per day, average 6 recipients - mix of HTML/Text/Attachments
- Read and Process Inbox messages 20 times per simulation day
- Load Attachments for 55% of messages. Average attachment size 1MB.
- Reply to: 0% of messages
- Forward 0% of messages
- Move 0% of messages
- Delete 18% of messages
- Request 1 meetings per day, average 1 attendees
- Respond to Meeting Requests 0%
- Respond to Meeting Requests 0%

Data size – The following table shows count and types of devices serviced by GMMS during the test(s):

Provisioned Good Mobile Messaging handhelds	1050
Handhelds running iOS	1050
Handhelds running Windows Mobile Pocket PC	0

Test Specifications

Testbed has a single GMC server with 2 GMM servers registered - MAPI and EWS/SQL GMM servers and was setup with 8 stress client hosts loaded with a set of users / batches as follows:

- Batch 1 - emulator 1 - 50 users
- Batch 2 - emulator 2 - 50 users
- Batch 3 - emulator 3 - 50 users

- Batch 4 - emulator 4 - 50 users
- Batch 5 - emulator 5 - 100 users
- Batch 6 - emulator 6 - 200 users
- Batch 7 - emulator 8 - 300 users

The following steps were performed for all user batches:

1. All users were first provisioned against MAPI GMM server, then brought online (connected/synced) with load specifications described above.
2. Load Generator was initiated to simulate external traffic to Exchange mailboxes.
3. Database size data point 1 was recorded.
4. Batch 1 was selected in GMC console and "Move user" function triggered.
5. Monitored GMMS log to record a time when First Time Sync is completed (100 email messages synced to the device)
6. Database size data point 2 was recorded.

Test Results

1. Sync time per stress client device seems to be constant with increasing number of users moved.

   50 user batch takes about 16-18 minutes, whereas 100 user move completes (with sync) in 32 minutes. 200 user batch took about 50 mins.

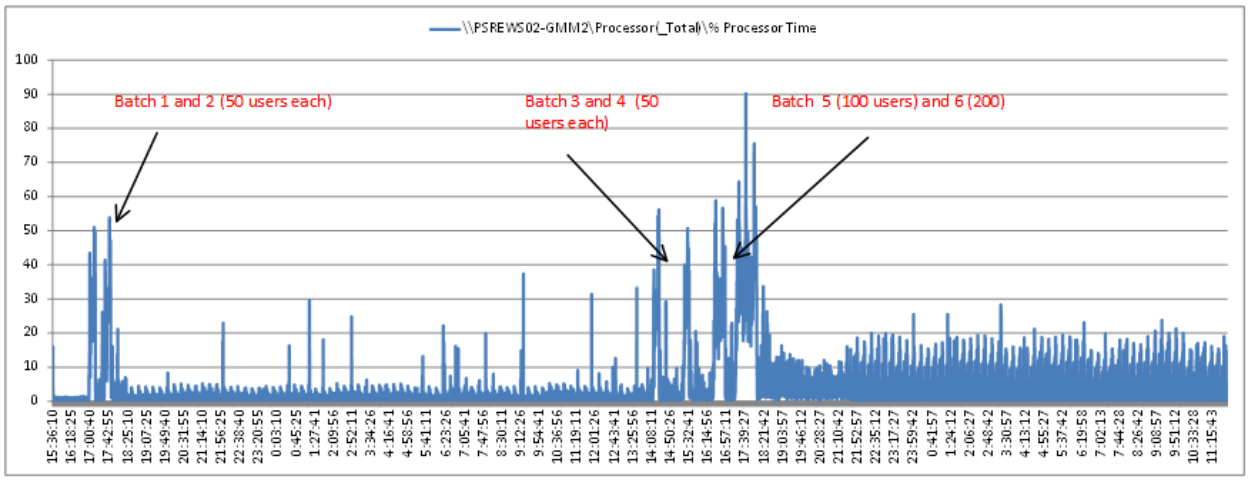
2. As number of moved users increases, the time to Resync an actual iOS device takes longer (stuck @ Starting Services for 1 hour) than Android (less than 10 mins).
3. Another issue found is where GMC doesn't allow selection of more than a 100 devices; hence, 200 users can't be moved at the same time.

   GMC Admin needs to schedule a move for 100 devices at a time or modify the GMC configuration to increase handheld page grid size to desired number.

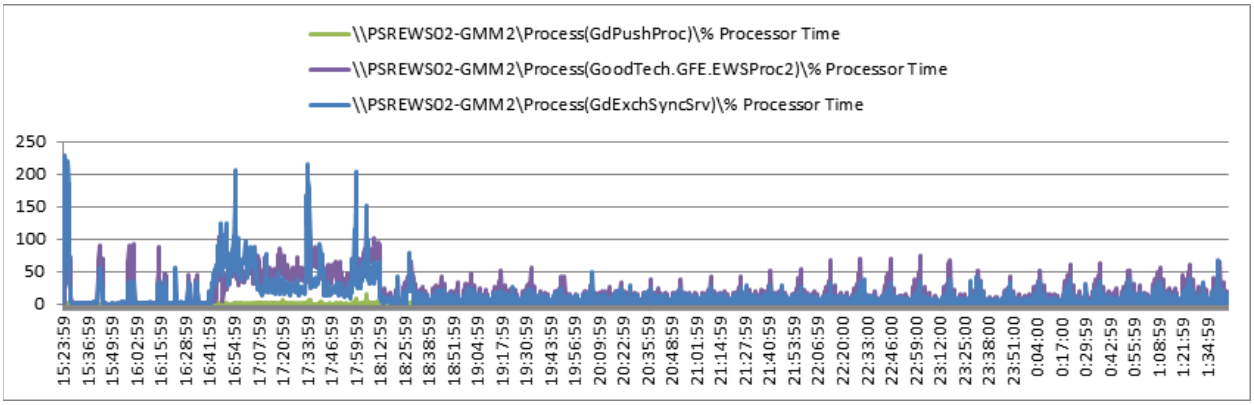
4. CPU usage during the move seemed normal. I'll follow up with a chart for the entire duration.

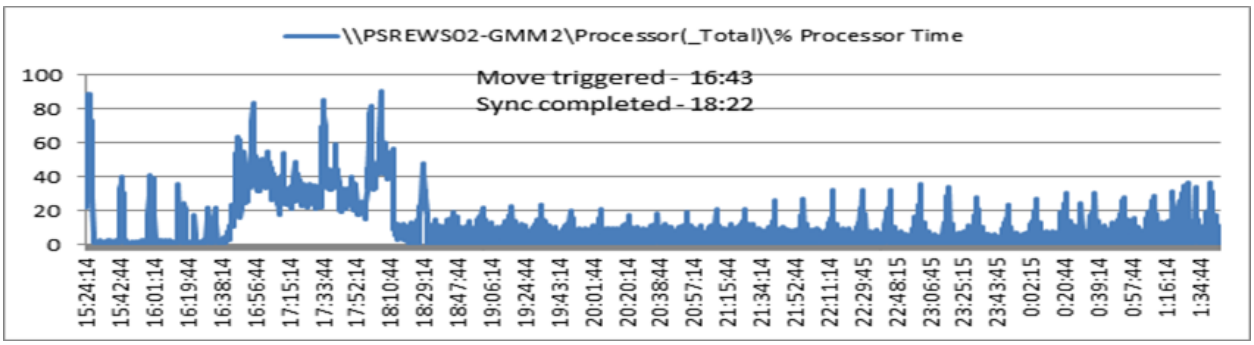
Batch 1 move triggered - completed	11/12/13 16:57 - 17:16
Batch 2	17:32 - 17:55
Batch 3	11/13/13 14:11 - 14:28
Batch 4	15:20 - 15:33
Batch 5	16:29 - 17:01
Batch 6	17:17 - 17:39
Batch 7	11/19/13 16:43 - 18:22

CPU utilization during moves 1 - 6



CPU utilization during move 7





Legal Notice

This document, as well as all accompanying documents for this product, is published by Good Technology Corporation ("Good"). Good may have patents or pending patent applications, trademarks, copyrights, and other intellectual property rights covering the subject matter in these documents. The furnishing of this, or any other document, does not in any way imply any license to these or other intellectual properties, except as expressly provided in written license agreements with Good. This document is for the use of licensed or authorized users only. No part of this document may be used, sold, reproduced, stored in a database or retrieval system or transmitted in any form or by any means, electronic or physical, for any purpose, other than the purchaser's authorized use without the express written permission of Good. Any unauthorized copying, distribution or disclosure of information is a violation of copyright laws.

While every effort has been made to ensure technical accuracy, information in this document is subject to change without notice and does not represent a commitment on the part of Good. The software described in this document is furnished under a license agreement or nondisclosure agreement. The software may be used or copied only in accordance with the terms of those written agreements.

The documentation provided is subject to change at Good's sole discretion without notice. It is your responsibility to utilize the most current documentation available. Good assumes no duty to update you, and therefore Good recommends that you check frequently for new versions. This documentation is provided "as is" and Good assumes no liability for the accuracy or completeness of the content. The content of this document may contain information regarding Good's future plans, including roadmaps and feature sets not yet available. It is stressed that this information is non-binding and Good creates no contractual obligation to deliver the features and functionality described herein, and expressly disclaims all theories of contract, detrimental reliance and/or promissory estoppel or similar theories.

Legal Information